Software and Systems

05 MAR 2012

Robert J. Bonneau, Ph.D.
Program Manager
AFOSR/RSL
Air Force Research Laboratory

Integrity ★ Service ★ Excellence

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
05 MAR 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Software And Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Air Force Research Laboratory,Wright Patterson AFB ,OH,45433**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the Air Force Office of Scientific Research (AFOSR) Spring Review Arlington, VA 5 through 9 March, 2012

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**20**	

2012 AFOSR SPRING REVIEW

NAME: Software and Systems

BRIEF DESCRIPTION OF PORTFOLIO:

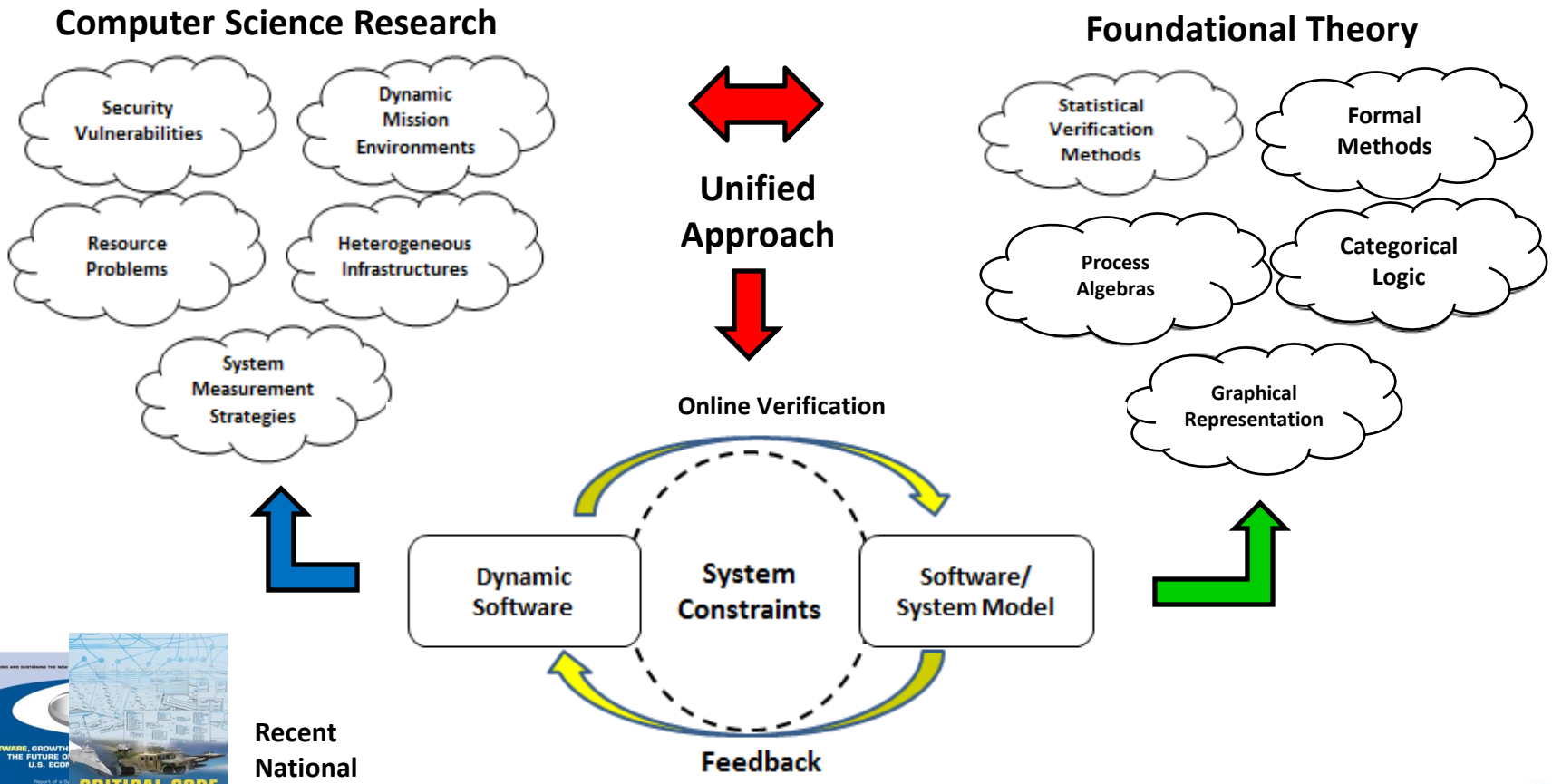
- **Enable quantifiable performance evaluation of critical software systems**
- **Manage software environments in order to preserve vital mission functions**
- **Comprehensively understand distributed effects in large software infrastructures to predict global system failures**

LIST SUB-AREAS IN PORTFOLIO:

- **Models for Composeable Dynamic Software**
- **Dynamic Formal Analysis and Verification**
- **Online Assessment and Repair of Failure**

Unified Approach to Software

- **Many current problems in software can be addressed in a more rigorous unified way by casting the software problem as a dynamic processes that can be measured and online management of software into existing and future systems**



Recent National Academies Reports

Current Program Scope

- **Models for Composeable Dynamic Software**
  - **New programming languages or language constructs reduce errors at run-time**
  - **Domain-specific languages enhance capabilities for code generation**
- **Dynamic Formal Analysis and Verification**
  - **Verification of system properties based on formal specifications**
- **Online Assessment and Repair of Failure**
  - **Abstract models of systems and their interactions facilitate automated generation of code**

Systems and Software

Agency Interaction

- OSTP/NITRD Coordinating Group
  - High Confidence Systems and Software (HCSS) Member
- ASDR&E
  - Software Producibility Initiative
- Secretary of the Air Force
  - Air Force Software and systems Overview Study
- NSF
  - Cyber Physical Systems
    - *Panelist and guest speaker at 2011 meeting*
- NASA
  - V&V of Flight Critical Systems
  - Ames Research Laboratory
    - Human Systems Integration Division
    - Intelligent Systems Division

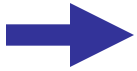
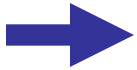
Systems and Software

Other funding agencies

- Army Research Office
  - Software investment mostly directed toward information assurance
- ONR
  - Software and Computing Systems
    - Principles for Correctness and Security Properties
  - Human Robot Interaction
  - Perception and Cognitive Control
- NSF
  - Cyber Physical Systems – focused on interaction with phyiscal environment and sensing systems
- DARPA: Software Producibility

Systems and Software

Program Trends

- Software Models Using Adaptive Feedback and Complexity Reduction 
- Feedback in Formal Analysis and Verification 
- Adaptive repair and assessment of distributed software infrastructures 
- Language-based approaches 
- Modeling Human-Machine Interaction 
- Agent-based approaches 

Software Contracts

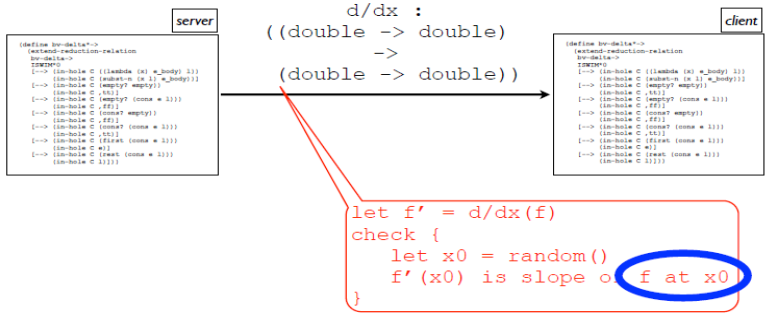
Felleisen, Northeastern

Approach: Software contracts incorporate feedback into models of online software assessment and require analysis of data type representation and meaning of data types to software performance

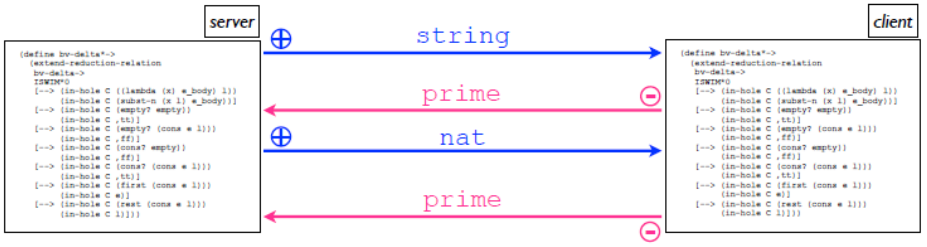
Payoff: Real time assessment of registers and data types in hardware software infrastructures can be performed

Contracts Create Ability To Trace Logical Errors

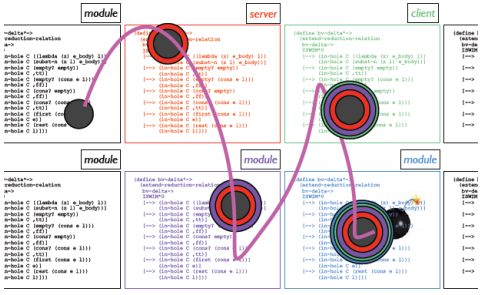
Trace of Logical Outcome of Mathematical Computation



Contract With Feedback



Contract + Feedback Allows Identification Of Logical Process Failure in Real Time



Scalable Model Checking

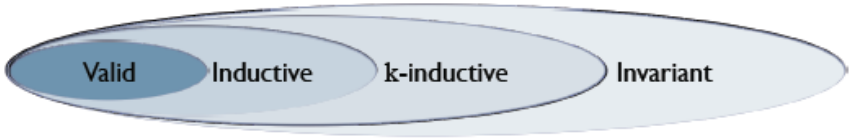
C. Tinelli U Iowa, C. Barret, NYU

Approach: Formal verification suffers from state space explosion. Compactly represent logical symbols in scalable nested satisfiability modulo theory (SMT)

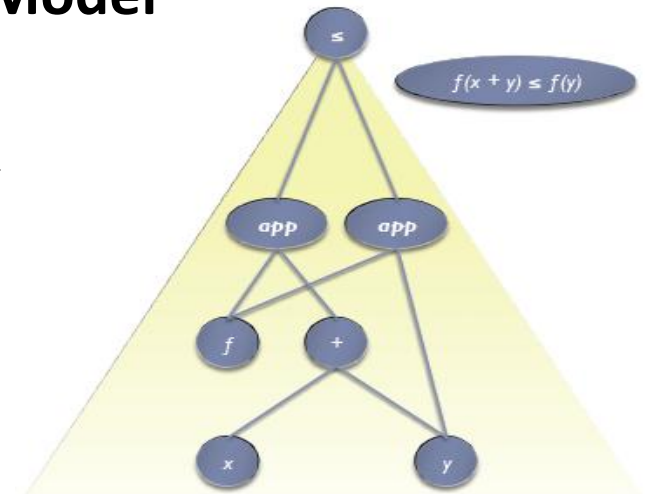
Payoff: More automated more scalable verification to handle large heterogeneous systems

Compact SMT Language

- Valid:
  - satisfied by all states in Q
- Inductive:
  - $I(s_0) \models P(s_0)$,
  - $P(s_n), T(s_n, s_{n+1}) \models P(s_{n+1})$
- k-inductive:
  - $I(s_0), T(s_0, s_1), \ldots, T(s_{k-1}, s_k) \models P(s_0), \ldots, P(s_k)$,
  - $T(s_n, s_{n+1}), \ldots, T(s_{n+k}, s_{n+k+1}), P(s_n), \ldots, P(s_{n+k}) \models P(s_{n+k+1})$
- Invariant:
  - satisfied by all reachable states of S



Improved Lower Dimensional Model



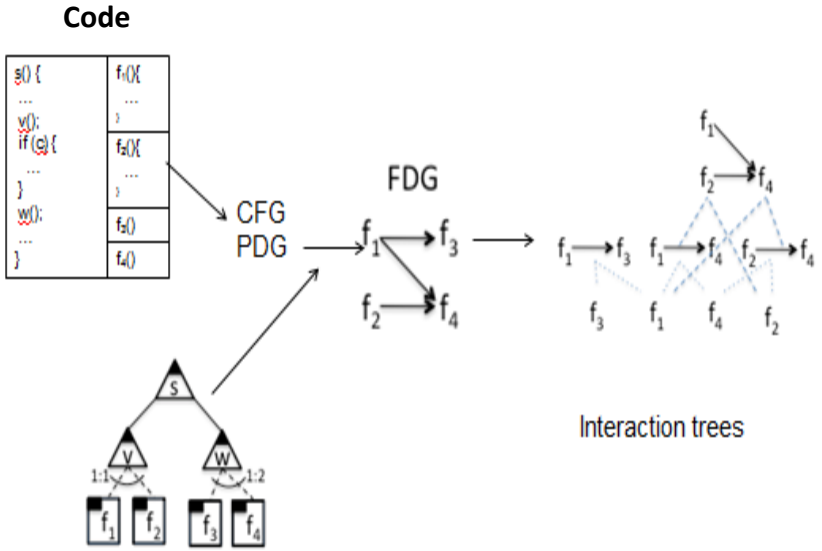
Adaptive Software Testing

Myra Cohen, U Nebraska Lincoln

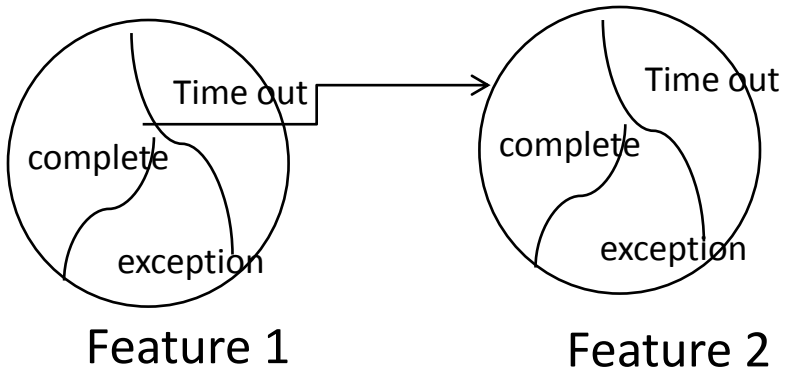
Approach: Understanding how to statistically represent a software model for software testing requires accurate models of mapping what to measure to performance

Payoff: Using a principled approach that captures the right level of software and abstraction statistically enables accurate statistical representation of failure modes

Statistical Software Testing and Measurement



Failure State Space Transitions Identified



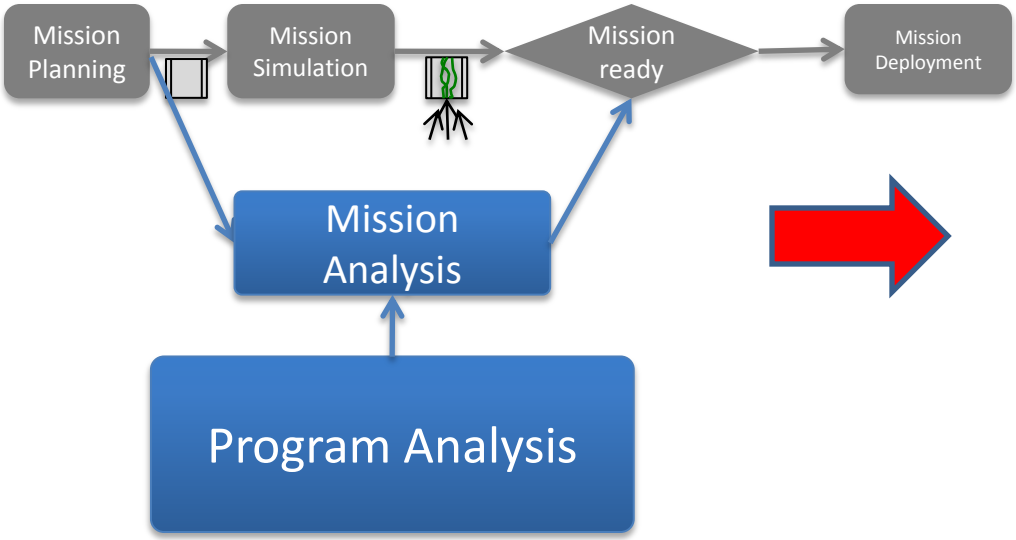
Mission Verification

Elbaum, Dwyer U. Neb., Rosenblum, U. Col. London

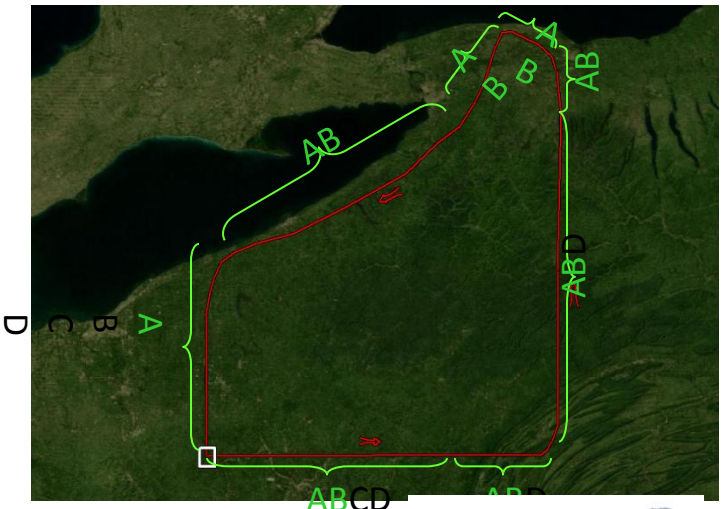
Approach: Develop a language to represent mission scenarios tied to integrated distributed software architecture.

Payoff: Verify global mission properties as function of lower level software constructs for quantifiable fault tolerance in achieving mission objectives

Mission Analysis Language Architecture



Fault Tolerant Mission Design



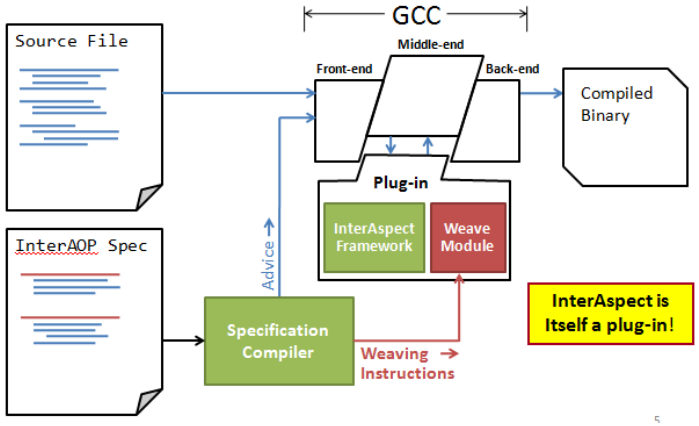
Feedback in Software Architecture

Smolka, Stony Brook, Havelund, JPL

Approach: Many software systems are introduced into environments that have uncertain conditions that result in unforeseen failures. Feedback failure correction mechanisms can augment software to adapt to failures

Payoff: Systems such as those on networks or those subject to uncertain physical environments can adapt to conditions using binary runtime repair of errors or faults based on automata theory and algebraic proofs of correctness

Robust Architecture with Feedback



Mathematical Formalism

Automata Description (feedback)

$$\mathrm{rt}(t) = \sum_{e \in O \cup \{\tau\}} \sum_{t'} \Delta_e^O(t, t').$$

Formal Logic (constraints)

$$P ::= X \mid \mathrm{nil} \mid a_{(w)}?t \mid b_{(r)}!t \mid \tau_{(r)} \cdot t \mid t_1 + t_2 \mid t_1 \, o_1 \|_{O_2} \, t_2 \mid t\,[O] \mid t\{a \leftarrow a'\} \mid \mu X.t$$

NASA Slated to Use Technology in Next Generation Mars Rover

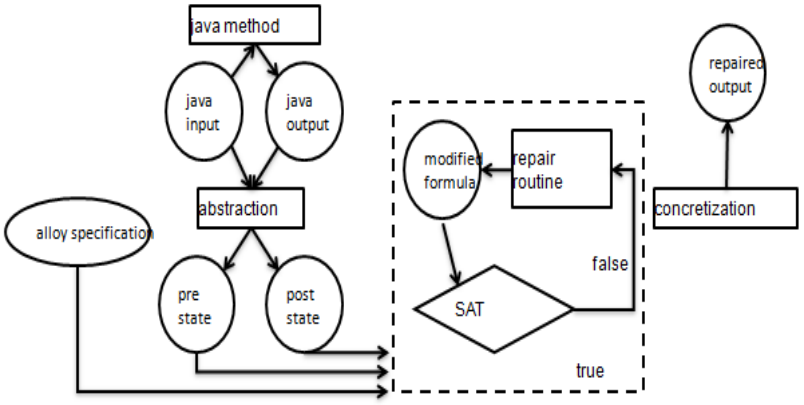
Runtime Repair

S. Khurshid, UT Austin

Approach: A functional approach can be developed for real time software runtime repair using new paradigms for online verification

Payoff: Faults in software can be corrected in real time and tracked rigorously

Real Time Runtime Software Repair Architecture



Results in Corrections of Multiple Faults

Measure	Alg.	List with 10 nodes					List with 20 nodes				
		Fail to fix the size	Delete another element too	Cycle	Broken List	Duplicate an element, wrong size	Fail to fix the size	Delete another element too	Cycle	Broken List	Duplicate an element, wrong size
Time (ms)	BM	7567	7314	8675	3809	5987	---	---	---	---	---
	IR	726	851	1038	911	1764	1061	9697	15784	9534	5744
	EL	614	13113	926	1160	1416	783	25581	18622	8852	4325
	GL	598	1060	870	1031	1136	807	14265	21180	13147	3522
Edit distance	BM	38	33	38	33	34	---	---	---	---	---
	IR	2	12	15	14	19	2	29	31	28	39
	EL	2	13	15	12	19	2	26	30	29	40
	GL	2	15	15	14	22	2	30	30	29	40

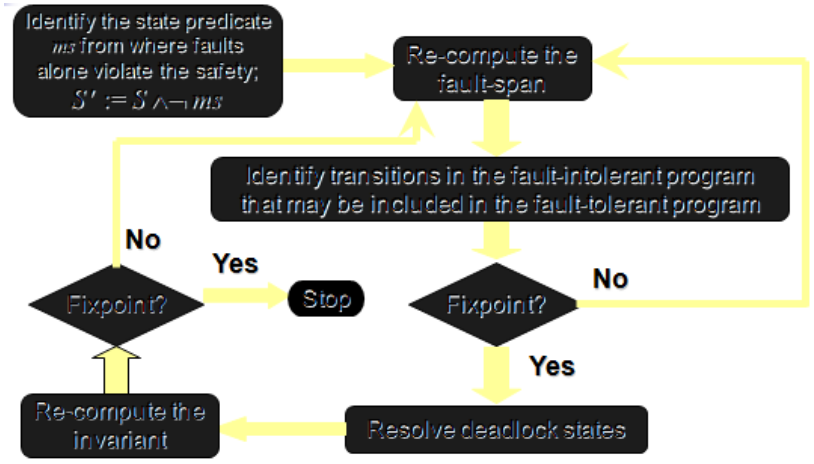
Automated Model Revision

Kulkarni, Mich State

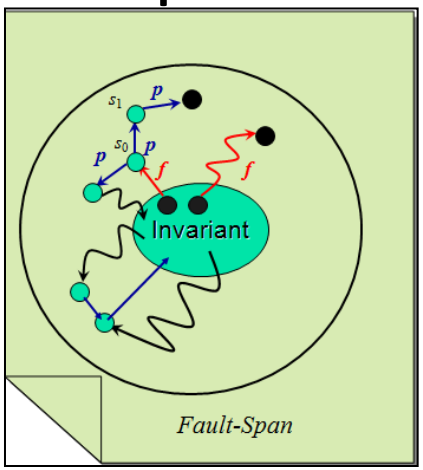
Approach: Verification tends to use approaches that are fixed based on the notion of pre-existing code and logical structures. In order to adapt to unanticipated conditions it is necessary to be able to revise models if conditions change.

Payoff: In dynamic heterogeneous systems, it is necessary to update the verification of the system as it evolves

Adaptive Verification



Adaptive Verification Space

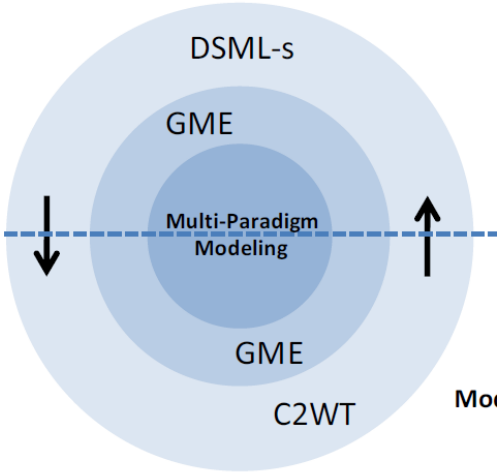


Question : Is it possible to *revise* the model automatically such that it satisfies the failed property while preserving the other propert

Systems and Software

AFRL Tech Directorate Interest/Coordination

- Information Directorate
  - Systems and Software Producibility
  - Multi-core Computing
- Air Vehicles
  - Flight-critical systems and software
  - Mixed-criticality architectures
- Human Effectiveness
  - Modeling of human-machine systems
  - Meta-information portrayal STTR
- Robust Decision Making STT
  - Large Scale Cognitive Modeling/C2WT



Increased Scale/Integration via DSMLs Anchored in DEVS (Douglass, 711th HPW/RH)

DEVS (*discrete event system specification*)

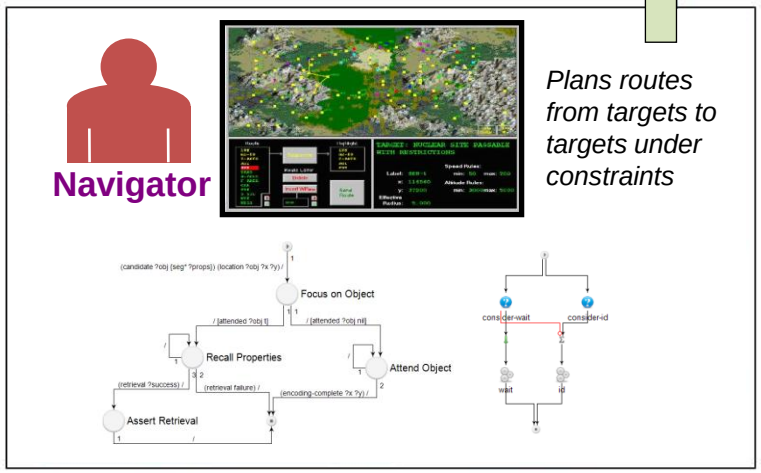
- Formal rigor
- Model reusability
- Interoperability

A discrete event system specification (DEVS) is a mathematical structure (7-tuple)

$$M = < X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta >$$

where

X	*is the set of input values*
S	*is a set of states*
Y	*is the set of output values*
$\delta_{int} : S \to S$	*is the internal transition function*
$\delta_{ext} : Q \times X \to S$	*is the external transition function*
$\lambda : S \to Y$	*is the output function*
$ta : S \to R_{0,\infty}$	*is the time advance function*



Plans routes from targets to targets under constraints

Domain-Specific Languages

- Tailored for cognitive modeling
- Semantically anchored in DEVS

High-Performance Computing

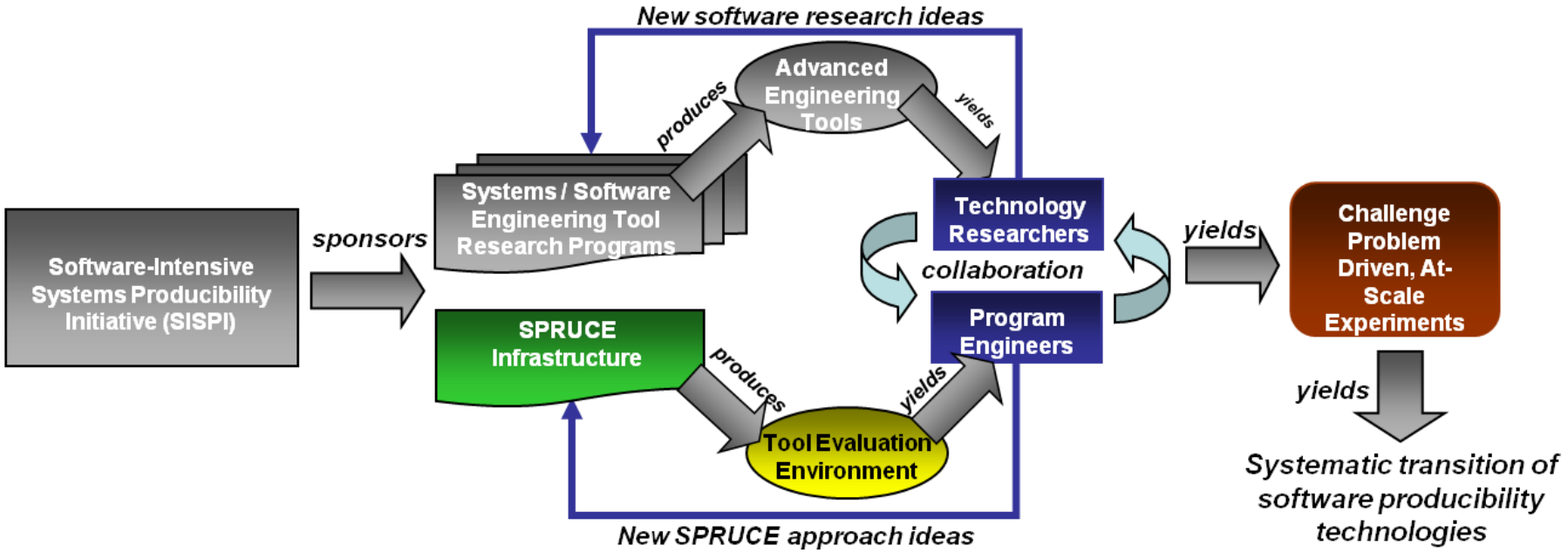
- Scalable simulation infrastructure
- Exploiting 25 years of DEV

SPRUCE

Drager/RI

Approach: Use parallel processing resources and network infrastructure as means of emulating and detecting system faults in new software deployment

Payoff: Deployment of new software tools has far fewer defects and more detailed assessment of integrated system performance



Software Collaborations at AFOSR

- **Information Operations and Security**
  - Fundamental software constructs for software and system security
- **Information Fusion**
  - Signal and sensor processing for integration of large data into systems architectures
- ***Complex Networks***
  - Mathematical and statistical methods for network and networked systems
- ***Foundations of Information Systems***
  - Measurement and statistical verification for software, network, and hardware
- **Computational Mathematics**
  - Methods of computational modeling of large complex physical processes
- **Dynamic Data Driven Applications Systems**
  - Strategies for real time feedback of data into distributed computational processes
- **Optimization and Discrete Mathematics**
  - Optimization strategies and algorithms for discrete computational processes
- ***Dynamics and Control***
  - Dynamical systems theory for assessment of performance of control architectures

Transitions

- Smolka/Havelund (Stony Brook/JPL)
  - *JPL Mars Science Laboratory* using rule-based specification language to ensure correct execution of software on next Mars Rover
- Harmonia STTR with AFRL/RI
  - using a modified version of Hadoop data analysis API for distributed parallel load balancing and computation over cloud architectures
- Tinelli/Barrett (Iowa/NYU)
  - *Rockwell-Collins* interested in transitioning SMT-based verifier research into formal methods toolkits for avionics systems
- Durfee (Univ of Michigan)
  - Collaboration on SBIR with *Intelligent Automation Inc.*, applying hybrid scheduling techniques to large-scale human expert teaming problems involving dozens of teams, hundreds of experts, and thousands of constraints.